

BİLEŞEN (DLL) İŞLEMLERİ

Programlardaki kod satırı sayısı günden güne artmaktadır. Bu sorunu çözmek için daha önce yazılmış olan kaynak kodu kullanarak daha hızlı uygulama geliştirmek üzere DLL yani bileşenler oluşturmak gerekmektedir. Buna ilave olarak önceden yazılmış kodları, bir şekilde (kaynak kodunu gizleyerek) hem başkasının kullanımına açmak hem de kendimizin tekrar kullanması amaçlanmıştır. Dll'ler tek başlarına çalışmazlar tek başlarına hiçbir işe yaramazlar ancak başka bir program tarafından kullanıldıklarında anlamlı olacaklardır. Yazılan bir dll'e istenilen bir programlama dilinden erişilebilir (tabi destekleyen diller). Örneğin C#'da yazılan bir dll'e Java'dan erişmek mümkündür. Böylelikle farklı dillerde kod yazan insanların ortak hareket noktası olarakta dll'ler kullanılabilirler.

Neden dll kullanmalıyız?

Dll kullanmalıyız çünkü tek bir dll yazarak bir çok sınıftan bu dll'e ulaşmamız mümkün olacaktır.

Yazılan Dll ram'e bir kere yüklenir ve orada kalır işlemiz olduğu zaman kullanırız işlemiz bitince dll'i ram'den atabiliriz (free etmek).

Bir program yazdığımızda bu programın alt programları aynı sınıfları kullanıyor ise her alt programı aynı fonksiyonu yazmaktansa tek bir dll yazarız ve tüm programlar bu dll'i kullanır yine bellekten avantaj sağlar.

Herhangi bir program yazıldığında bu programa güncellemelerin geleceği aşıkardır, bundan ötürü program komple güncelleneceğine tek bir dll dosyası güncellenir ve bu bize hız sağlar komple bir programın güncelleneceği yere tek bir dll güncellemek daha kolay ve hızlı olacaktır.

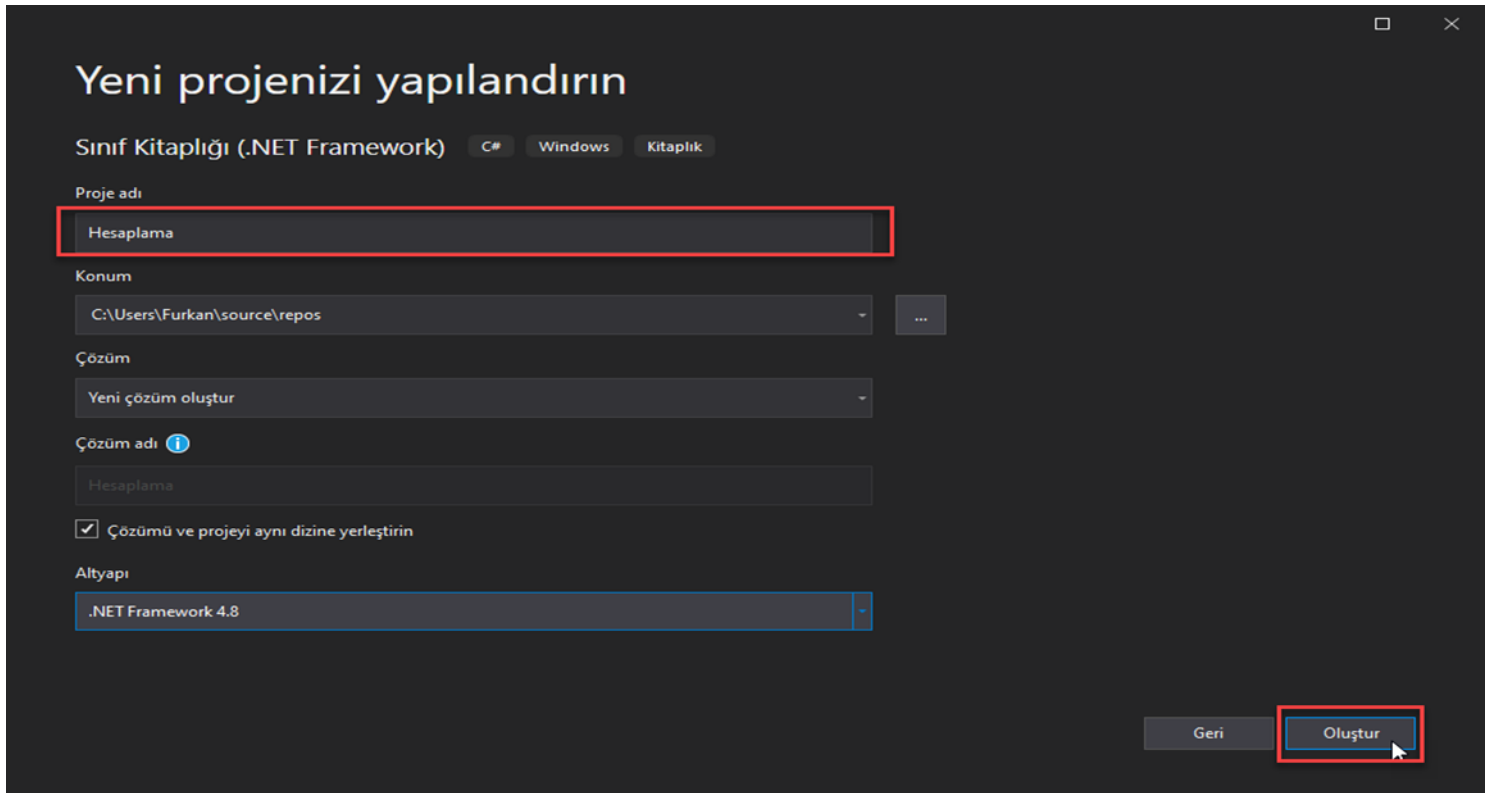
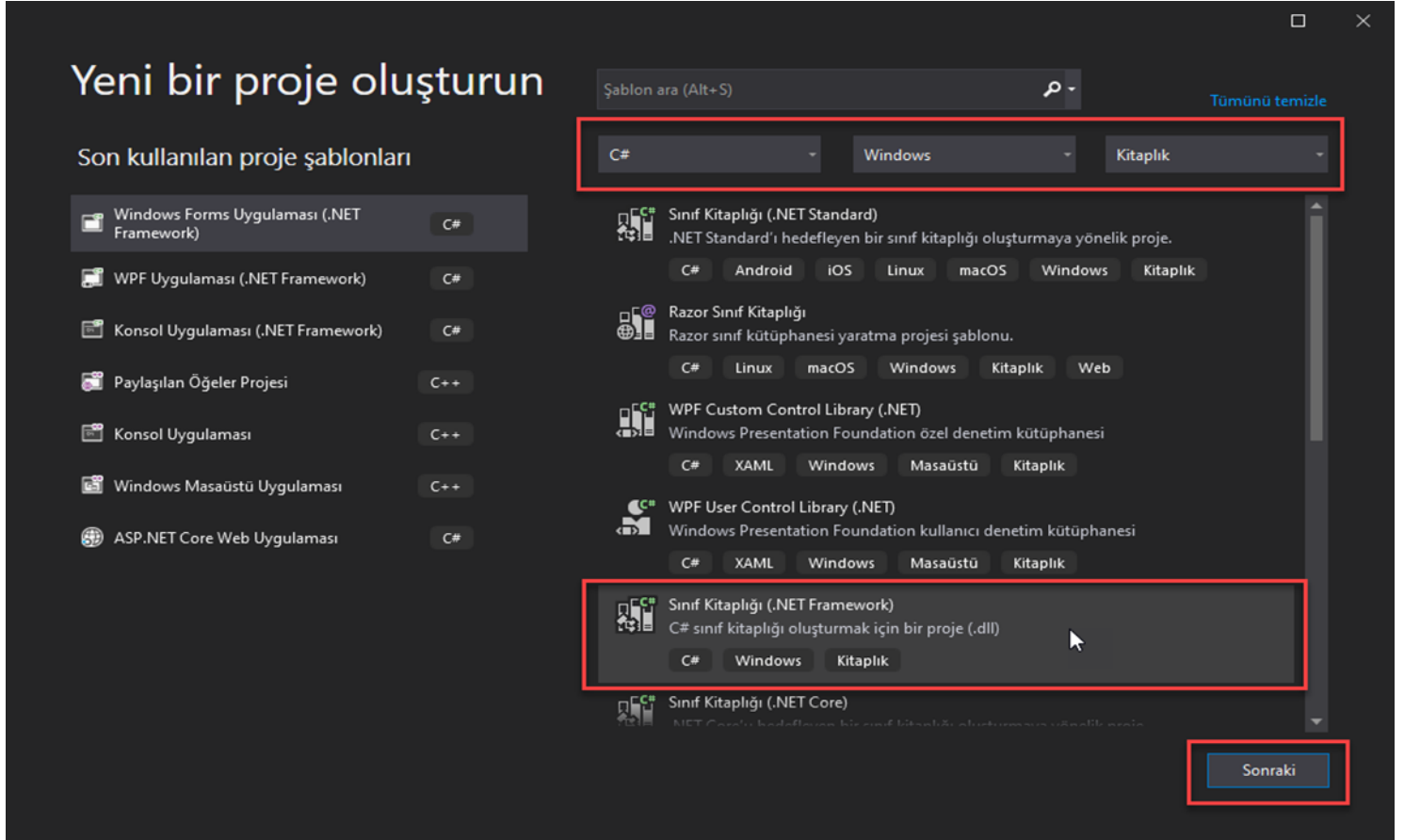
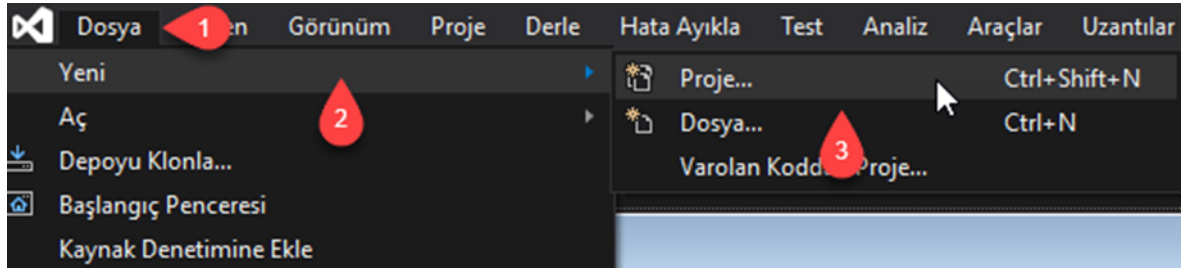
Statik ve dinamik dll arasındaki farklar;

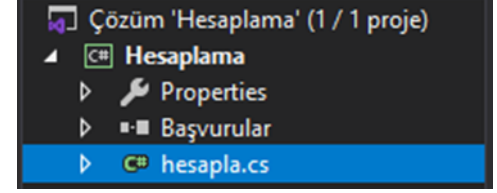
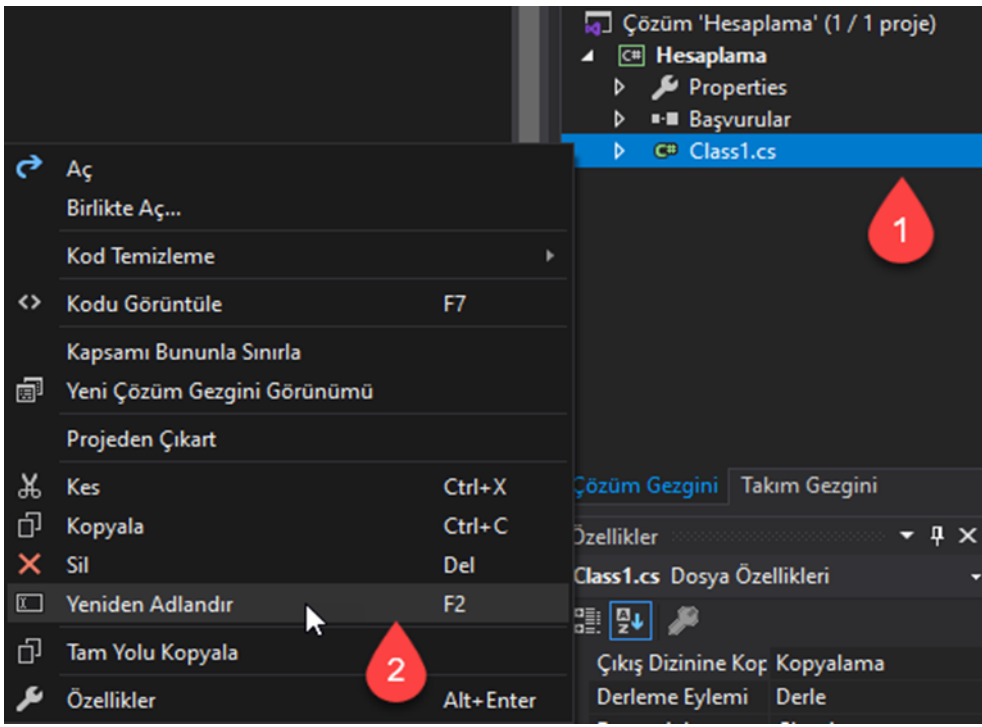
Aslında statik ve dinamik dll olayı dll'i projemiz nasıl dahil ettiğimiz ile alakalıdır.

Statik Dll : Bu dll türünde dll el ile yolu gösterilerek çok kolay bir şekilde projeye dahil edilir. Eklenen dll statik eklendiği için kullanılsada kullanılmasada projede vardır ve ram'de yer işgal eder bu durum aslında statik yüklemenin en büyük dezavantajıdır. Statik yüklemde eğer bir şekilde dll bulunamaz ise program crash olur ve hata verir bu ikincisi en büyük dezavantajdır. Bu dezavantajların dışındaki tüm durumlarda statik yüklene dll yukarıda bahsettiğimiz tüm avantajları sağlamaktadır.

Dinamik Dll : Profesyonel kullanıcılar için daha iyidir, statik yüklemenin getirdiği tüm dez avantajları yok eder dll ile işlemiz olduğunda gelir işlemiz bitince ram'den silinir gereksiz yer işgal etmez. Program dll'i bulamadığı zaman çalışmasına devam eder crash olmaz.

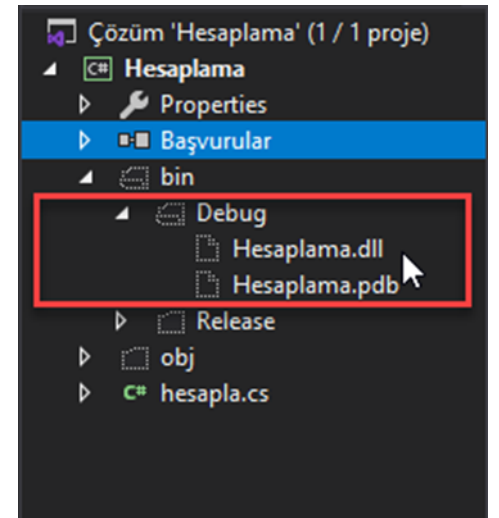
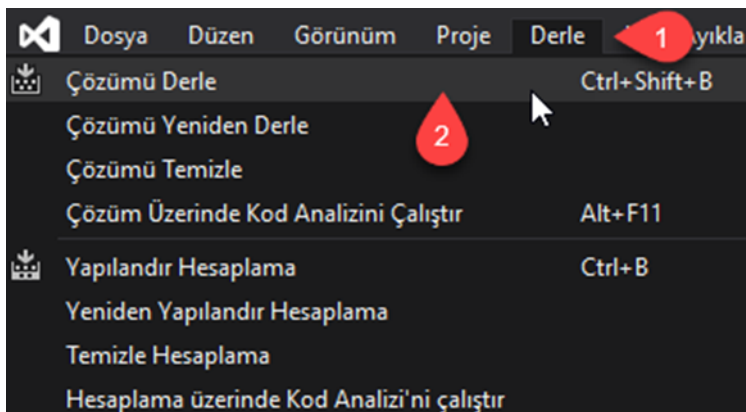
Bileşen Oluşturmak





```
namespace Hesaplama
{
    0 başvuru
    public class hesapla
    {
        0 başvuru
        public int Topla(int a, int b)
        {
            return a + b;
        }
        0 başvuru
        public int Cikar(int a, int b)
        {
            return a - b;
        }

        0 başvuru
        public int Carp(int a, int b)
        {
            return a * b;
        }
        0 başvuru
        public int Bol(int a, int b)
        {
            return a / b;
        }
    }
}
```



Bilesen Kullanmak

Hesapla

İlk Sayı Topla Çıkar

İkinci Sayı Çarp Böl

Sonuç

Hesapla

İlk Sayı Topla Çıkar

İkinci Sayı Çarp Böl

Sonuç

Yeni Öğe... Ctrl+Shift+A
Var Olan Öğe... Shift+Alt+A
Yeni Klasör
REST API İstemcisi...
Başvuru...
Web Başvurusu...
Hizmet Başvurusu...
Bağlı Hizmet
Çözümleyici...
Form (Windows Forms)...
Kullanıcı Denetimi (Windows Forms)...
Bileşen...
Sınıf...
Yeni EditorConfig

Derle
Tekrar Derle
Temizle
Çözümleme ve Kod Temizleme
Yayımla...
Kapsamı Bununla Sınırla
Yeni Çözüm Gezgini Görünümü
Ekle
NuGet Paketlerini Yönet...
Başlangıç Projesi Olarak Ayarla
Hata Ayıkla
Projeyi Etkileşimli Pencerede Başlat
Kaynak Denetimi
Kes Ctrl+X
Yapıştır Ctrl+V
Kaldır Del
Yeniden Adlandır F2
Projeyi Kaldır
Projenin Doğrudan Bağımlılıklarını Yükle
Projenin Tüm Bağımlılık Ağacını Yükle
Tam Yolu Kopyala
Klasörü Dosya Gezgini'nde Aç
Open in Terminal
Özellikler Alt+Enter

Çözüm Gezgini İçinde Ara (Ctrl+Ş)
Çözüm 'Database' (1 / 1 proje)
Database
Properties
Başvurular
SqlServerTypes
App.config
Database2.cs
DataSet1.xsd
DataSet2.xsd
Form1.cs
Form10.cs
Form11.cs
Form12.cs
Form13.cs
Çözüm Gezgini Takım Gezgini
Özellikler
Database Proje Özellikleri
Proje Dosyası Database.csproj
Proje Klasörü C:\Users\Furkan\source\repos\Hesapla\Hesapla\bin\Debug\Hesapla.exe

Hata Listesi
Tüm Çözüm 0 Hata 0 Uyarı
Kod Açıklama Proje

Başvuru Yöneticisi - Database

Derlemeler

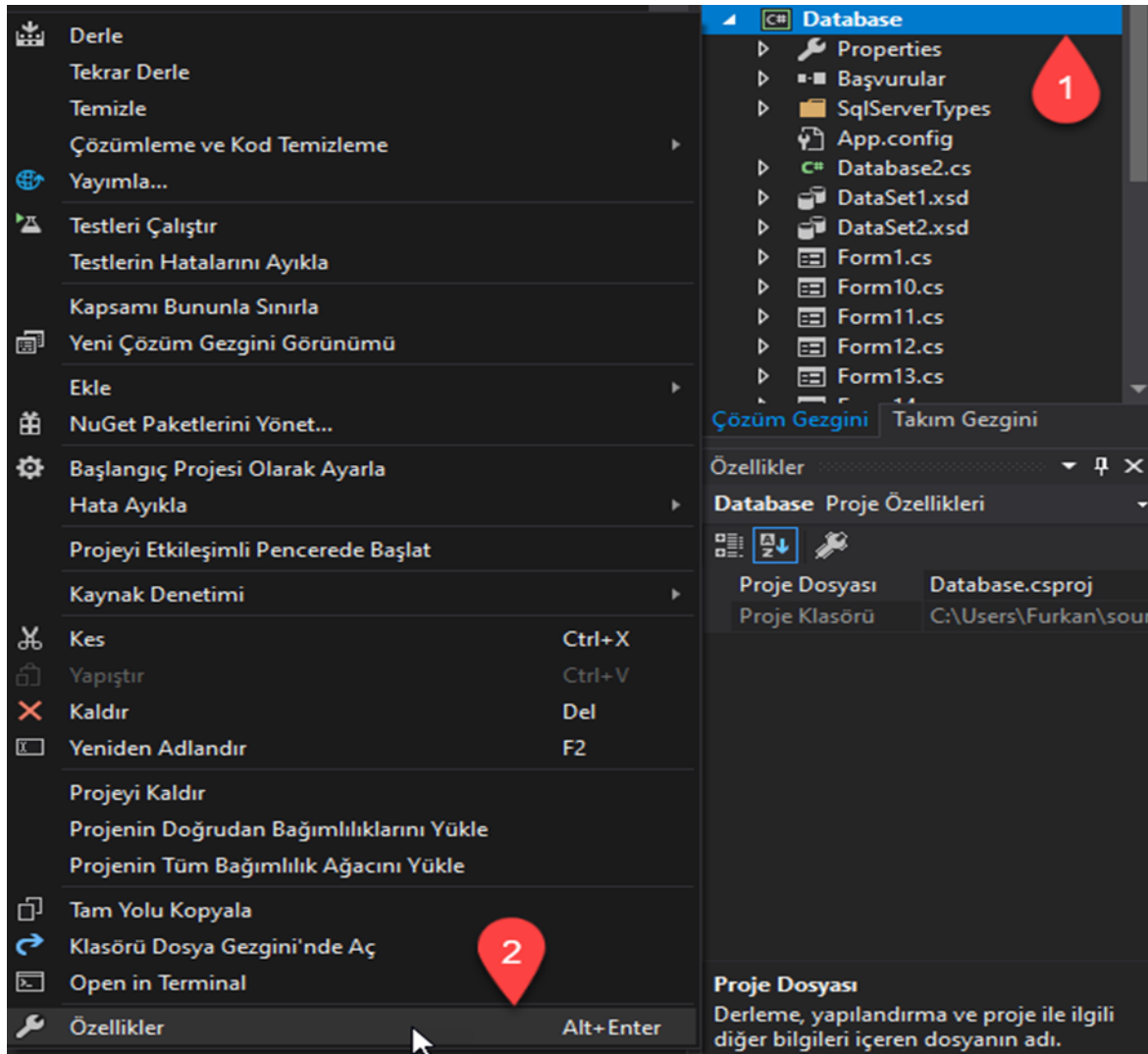
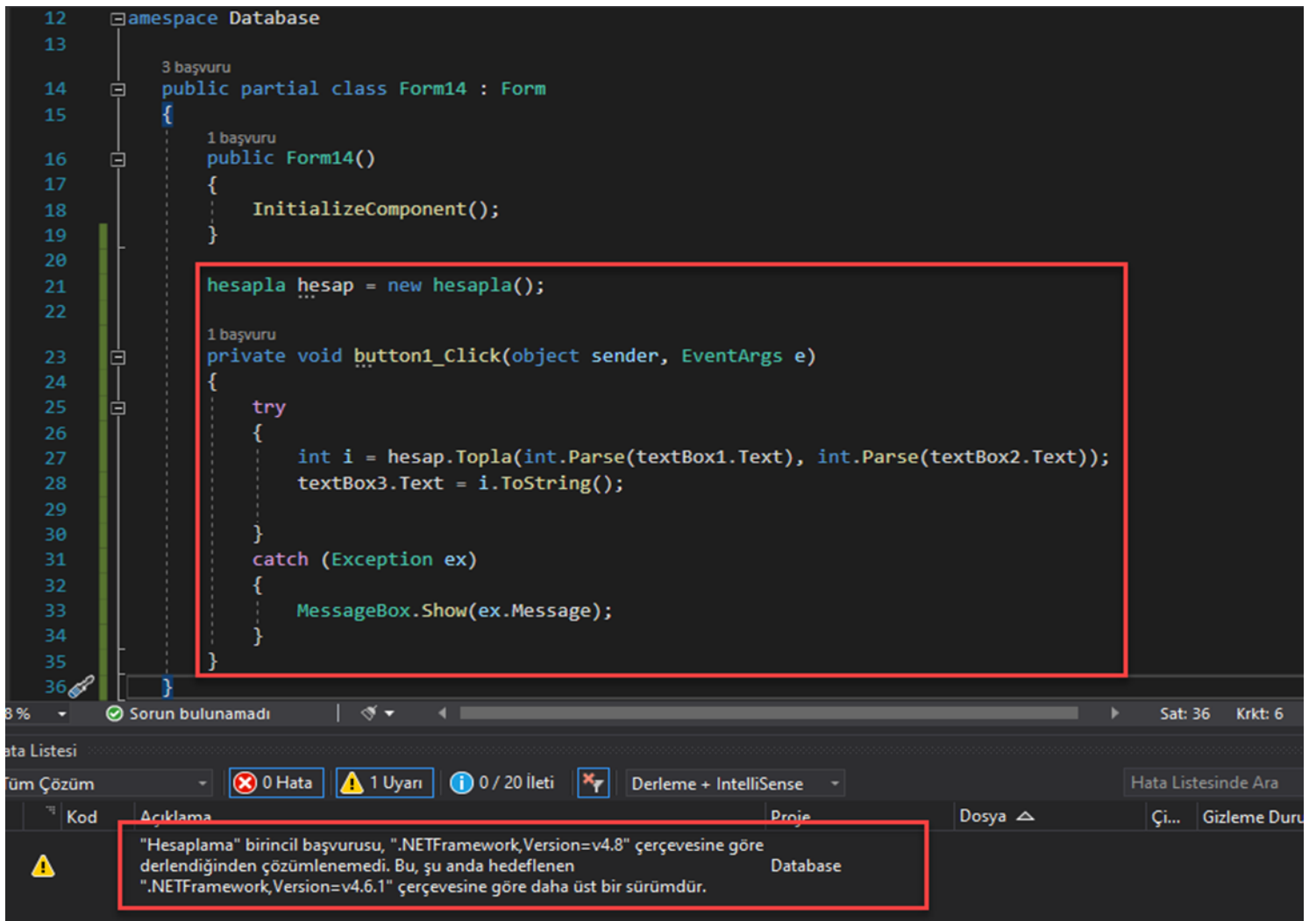
Hedeflenen: .NET Framework 4.6.1

Ara (Ctrl+E)

Ad	Sürüm
Accessibility	4.0.0.0
CustomMarshalers	4.0.0.0
ISymWrapper	4.0.0.0
Microsoft.Activities.Build	4.0.0.0
Microsoft.Build	4.0.0.0
Microsoft.Build.Conversion.v4.0	4.0.0.0
Microsoft.Build.Engine	4.0.0.0
Microsoft.Build.Framework	4.0.0.0
Microsoft.Build.Tasks.v4.0	4.0.0.0
Microsoft.Build.Utilities.v4.0	4.0.0.0
Microsoft.CSharp	4.0.0.0
Microsoft.JScript	10.0.0.0
Microsoft.VisualBasic	10.0.0.0
Microsoft.VisualBasic.Compatibility	10.0.0.0
Microsoft.VisualBasic.Compatibility.Data	10.0.0.0
Microsoft.VisualBasic	10.0.0.0
Microsoft.VisualBasic.STLCLR	2.0.0.0
mscorlib	4.0.0.0
PresentationBuildTasks	4.0.0.0
PresentationCore	4.0.0.0
PresentationFramework	4.0.0.0
PresentationFramework.Aero	4.0.0.0
PresentationFramework.Aero2	4.0.0.0
PresentationFramework.AeroLite	4.0.0.0
PresentationFramework.Classic	4.0.0.0
PresentationFramework.Luna	4.0.0.0

Ad: Accessibility
Oluşturan: Microsoft Corporation
Sürüm: 4.0.0.0
Dosya Sürümü: 4.6.1055.0 built by: NETFXREL2

Gözet... Tamam İptal



Uygulama

Derleme

Derleme Olayları

Hata Ayıkla

Kaynaklar

Hizmetler

Ayarlar

Başvuru Yolları

İmzalama

Güvenlik

Yayımlama

Kod Analizi

Yapılandırma: Yok Platform: Yok

Bütünleştirilmiş kod adı: Database Varsayılan ad alanı: Database

Hedef Framework: .NET Framework 4.6.1 Çıkış türü: Windows Uygulaması

☒ Otomatik olarak bağlama yeniden yönlendirmeleri oluştur

Başlangıç nesnesi: (Ayarlanmadı) Bütünleştirilmiş Kod Bilgileri...

Kaynaklar

Uygulama kaynaklarının nasıl yönetileceğini belirtin:

☒ Simge ve bildirim

Bildirim, bir uygulamaya özgü ayarları belirler. Özel bir bildirim eklemek için önce bildirimi projenize ekleyin, sonra aşağıdaki listeden seçin.

Simge: (Varsayılan Simge) Gözet...

Bildirim: Bildirimi varsayılan ayarlarla ekle

☐ Kaynak dosyası: Gözet...

Uygulama

Derleme

Derleme Olayları

Hata Ayıkla

Kaynaklar

Hizmetler

Ayarlar

Başvuru Yolları

İmzalama

Güvenlik

Yayımlama

Kod Analizi

Yapılandırma: Yok Platform: Yok

Bütünleştirilmiş kod adı: Database Varsayılan ad alanı: Database

Hedef Framework: .NET Framework 4.8 Çıkış türü: Windows Uygulaması

☒ Otomatik olarak bağlama yeniden yönlendirmeleri oluştur

Başlangıç nesnesi: (Ayarlanmadı) Bütünleştirilmiş Kod Bilgileri...

Kaynaklar

Uygulama kaynaklarının nasıl yönetileceğini belirtin:

☒ Simge ve bildirim

Bildirim, bir uygulamaya özgü ayarları belirler. Özel bir bildirim eklemek için önce bildirimi projenize ekleyin, sonra aşağıdaki listeden seçin.

Simge: (Varsayılan Simge) Gözet...

Bildirim: Bildirimi varsayılan ayarlarla ekle

☐ Kaynak dosyası: Gözet...

Hedef Framework Değişimi

Hedef Çerçevenin değiştirilmesi geçerli projenin kapatılmasını ve daha sonra yeniden açılmasını gerektirir. Proje içerisindeki kaydedilmemiş değişiklikler otomatik olarak kaydedilecektir.

Hedef Çerçevenin değiştirilmesi, proje dosyalarının derlenebilmeleri için el ile değiştirilmelerini gerektirebilir.

Bu proje için Hedef Çerçeveyi değiştirmek istediğinize emin misiniz?

Evet Hayır Yardım

Hata Listesi

Tüm Çözüm 0 Hata 0 Uyarı 0 İleti Derleme + IntelliSense Hata Listesinde Ara

Kod	Açıklama	Proje	Dosya	Çi...	Gizleme Durumu